Employee Management System (ASP.NET Core Razor Pages + PostgreSQL)

================================================================

1. Project Setup:

-----------------

- ASP.NET Core Razor Pages (.NET 8)
- PostgreSQL (via Npgsql.EntityFrameworkCore.PostgreSQL)
- Configured EF Core (Code-First)
- Connection String in appsettings.json

2. Database:

------------

- AspNetUsers (Identity Users)
- Employees Table: Id, Name, Position, Email, DateOfJoining, Salary, ProfilePicturePath, UserId, DocumentPath

3. Roles:

---------

- Admin: Manages all Employees (CRUD + Documents)
- Employee: Manages own MyDetails & MyProfile

4. Features:

------------

Employee:

- Register/Login

- View/Update MyDetails
- Upload Profile Picture (MyProfile)
- Read-only profile picture on MyDetails

Admin:

- View All Employees
- Create/Edit/Delete Employee
- Upload/View Document for Employees
- Document replaces old on re-upload

5. UI/UX:

---------

- Bootstrap for styling
- Medium-sized input boxes
- Profile picture placed well
- Toast (snackbar) on document upload

6. Exception Handling:

----------------------

- Try-catch in file uploads
- ModelState.IsValid for form checks

7. Home Redirects:

------------------

- Admin: /Employees/Index
- Employee: /Employees/MyDetails

8. Git:

-------

- Pushed to Private GitHub Repo

- Cloning via HTTPS link

- PostgreSQL DB Dump via pg_dump

9. Architecture:

----------------

- Monolithic Layered Architecture

- Presentation: Razor Pages

- Business Logic: PageModel (.cshtml.cs)

- Data Access: EF Core + PostgreSQL

10. PostgreSQL Backup:

----------------------

- pg_dump used for DB export

- Can be restored via psql or pgAdmin

Prepared for Interview & Practical Use.